Unix/X-windows meets the Etherswitch

Tony Dale,
Computer Science Dept,
University of Canterbury

March 17, 1995

1 Those overloaded network blues

You know the feeling — happily typing/mousing away on your Xterminal/workstation/whatever, suddenly your mouse cursor seems stuck in treacle, your typing echoes in fits and starts, and deiconifying that Console window takes an age. You've got the low–down, slow–down, overloaded ethernet blues.

In todays typical networking environment, an overloaded ethernet can be quite a problem to fix. Often an ethernet operates just at the "knee" of the ethernet response curve (see figure 1). A sudden heavy demand for bandwidth, such as from a heavily–paging workstation, can cause a dramatic rise in network response time.

If your Unix network is like ours it will be a somewhat hodge-podge collection of file/compute servers with various workstations and Xterminals hooked up via a collection of ethernet repeaters and bridges. The network topology may have been decided by what was most expedient at the time of installation. There may be lots of filesharing traffic travelling over the ethernet, fighting with X terminal traffic, virtual–memory paging traffic, etc. All this adds up to a network which may not respond well to traditional subnetting solutions.

2 Subnetting: the traditional fix

Traditionally the fix for Unix network overloads has been to divide an ethernet into several subnets, using protocol-dependent routers. Each subnet would have its own IP network number, which would identify all the nodes on the subnet. An IP gateway, either a dedicated IP router or a Unix box with two ethernet interfaces, would separate the subnet from other subnets. Machines on other subnets need to know a route (the address of a gateway) to the new subnet, and vice–versa. Obviously, such an approach is very difficult to change, so I will call it "static subnetting".

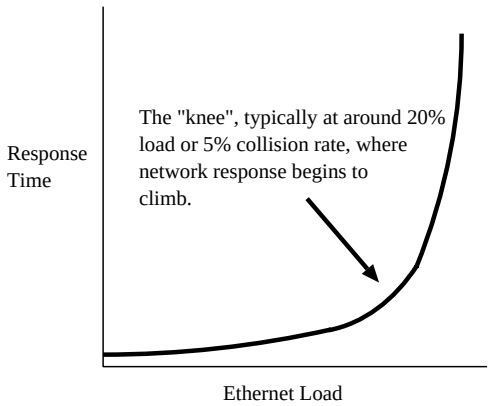

Figure 1: Ethernet response time v's load.

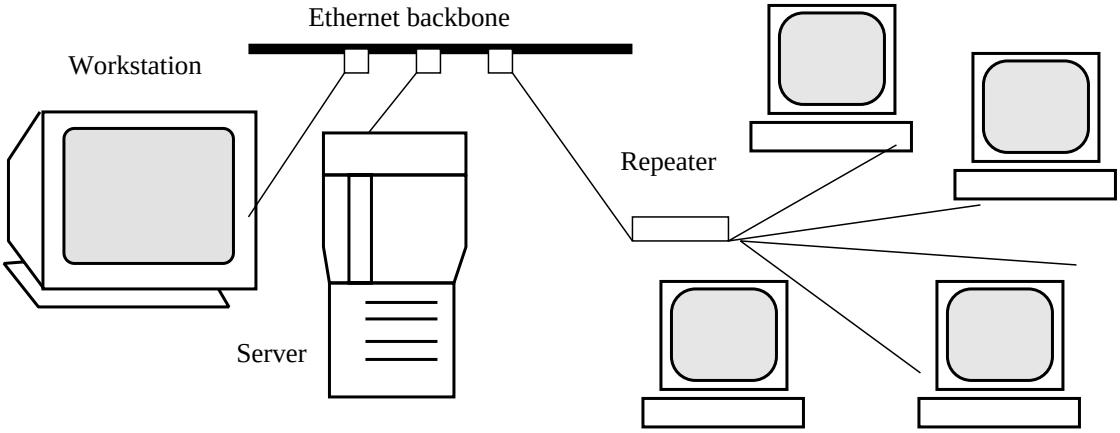

Figure 2: A simple network.

The characteristics of static subnetting are:

1. All connections and routes are unchanging.
2. It may use protocol routers, or ethernet-level bridges. Bridges are easier to set up (just plug them in and turn them on) but they provide poorer isolation between subnets. For example, bridges have to pass ethernet broadcasts through to all subnets but routers generally do not.
3. Packet forwarding may be quite slow. There may be a maximum forwarding rate of less than 10,000 packets/second through a slow router.
4. It may be necessary to manually administer routing information in Protocol – level routers and/or hosts, eg: for static IP routing.
5. Subnets contain relatively large numbers of nodes. Routers are used to isolate traffic between subnets, not to forward large quantities of packets.

Static subnetting can work very well, providing relief from network overloads, but it requires careful planning. For instance, suppose that we have a situation with a file/compute server, a fast workstation which shares files from the server and a cluster of Xterminals, all on one network, see figure 2.

Some users find out that the fast workstation runs their Xsessions faster than the central file/compute server. Pretty soon lots of them are running their sessions from the workstation, generating lots of filesharing traffic and suddenly system response becomes diabolical, with a network load of 50%.

Our hero, the system administrator (part – time) looks at the network load generated from the workstation (high), the central file/compute server (high), and the Xterminals (moderate). A bit of head – scratching and an educated guess, and the decision is made to add an ethernet card to the central fileserver, splitting the network into two subnets, as in figure 3. The same effect could be had by putting a two – port bridge between the Xterminals and the central server and workstation.

The fast workstation goes on one subnet and the Xterminals go on the other subnet. That ought to cut the network traffic down to about half on one network and half on the other, right? Well, no. What actually happens is that the load on the Xterminal network goes right down, but the workstation/server network stays high.

This is because the Xterminal users are still thrashing that workstation. Their traffic is a minor load, but it generates lots of filesharing traffic on the Server — Workstation subnet, and this competes with the X traffic.

Maybe the Xterminal users will switch back to the central server. Maybe they will be banned from the fast workstation. Whatever the case, subnetting did not fix the ethernet overload. It may actually make performance quite a bit worse because the servers IO bus could be overloaded by having to route all the packets from the Xterminals, although that would be less of a problem with an external router.

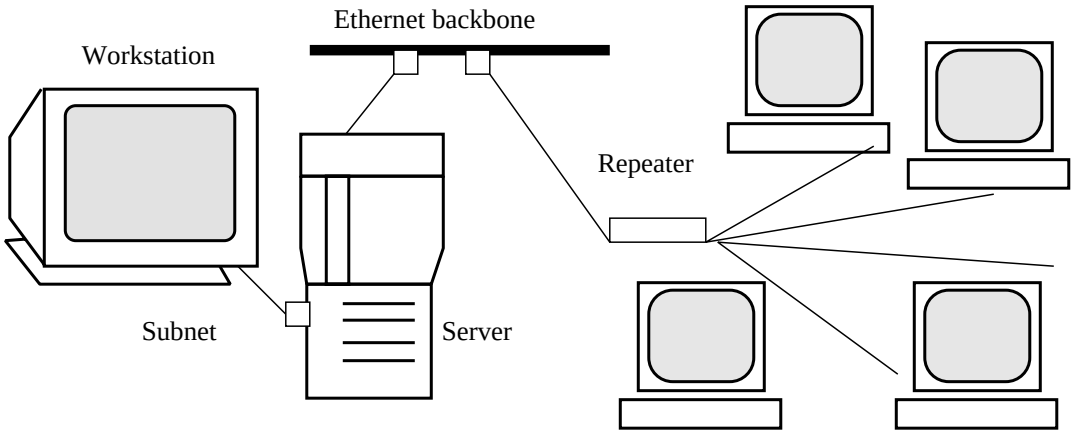

Figure 3: A simple subnetted network.

3 Dynamically switching ethernet hubs

Now becoming popular, dynamically switching ethernet hubs provide an effective and, in some cases, a cheap way of subnetting a heavily loaded ethernet. These hubs constantly change their subnet configuration according to the packets arriving at their ports. Ideally, every port receiving a packet is instantly connected to the port with the destination address for that packet.

The characteristics of a dynamically switching ethernet hub are:

1. Connections and routes change with each new packet arriving.
2. Many simultaneous ethernet "conversations" happen at once.
3. Works at the ethernet protocol layer, and so for that reason is like an ethernet bridge rather than a protocol – dependant router. Eg: ethernet broadcasts are passed through to all subnets.
4. Very high performance, able to forward a continuous stream of packets.
5. Subnets may consist of only one node, consequently routers may have to forward large amounts of traffic.

In the most extremely subnetted case there is only one workstation/Server, etc, per hub port, and each port provides what amounts to a 10 Mbps subnet. We found that major gains in network response can be made with just two or three dynamically – switched "subnets", if you are careful to find sources of heavy network load.

How might a dynamic hub help the network of the previous section? Well, let's put it between the fast workstation, the server and the Xterminals, as in figure 5.

Now the ethernet loads go down, with moderate traffic between the workstation and the server, and a low load on the Xterminal network. Why is that? Let's look at a real example for the explanation. . .

4 A real example

The network in our Computer Science department has grown over the years, and its chequered history shows in the topology of the network, see figure 6.

As can be seen, there are a three subnets, the first one was to isolate our ethernet from the campus backbone traffic, the second two were subnets to help cut down ethernet load in our departmental network.

We noticed a significant reduction in load — down from 20% to 10% when we isolated ourselves from the campus ethernet backbone, and likewise at the when time we subnetted our departmental network loads were cut down from 20% to 10% on one subnet and 5% on the other. This was because there were about a dozen workstations and compute servers, with perhaps a half-dozen Xterminals scattered around. Most of the traffic was NFS filesharing traffic, and so adding a second ethernet

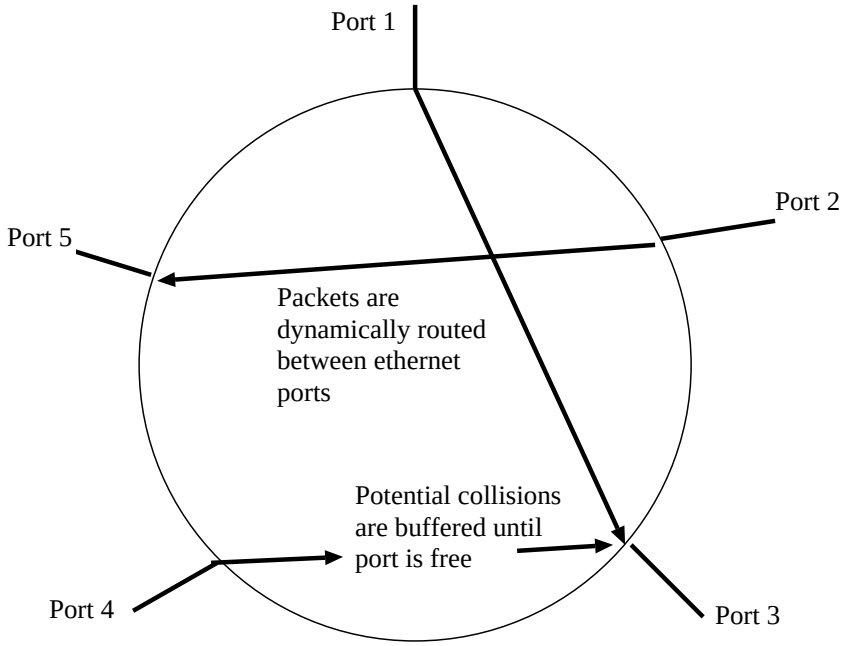

Figure 4: Conceptual diagram of a dynamic ethernet hub. There is a time delay involved in switching the packets, ranging from 40 microseconds to several milliseconds depending on the switching technology used.

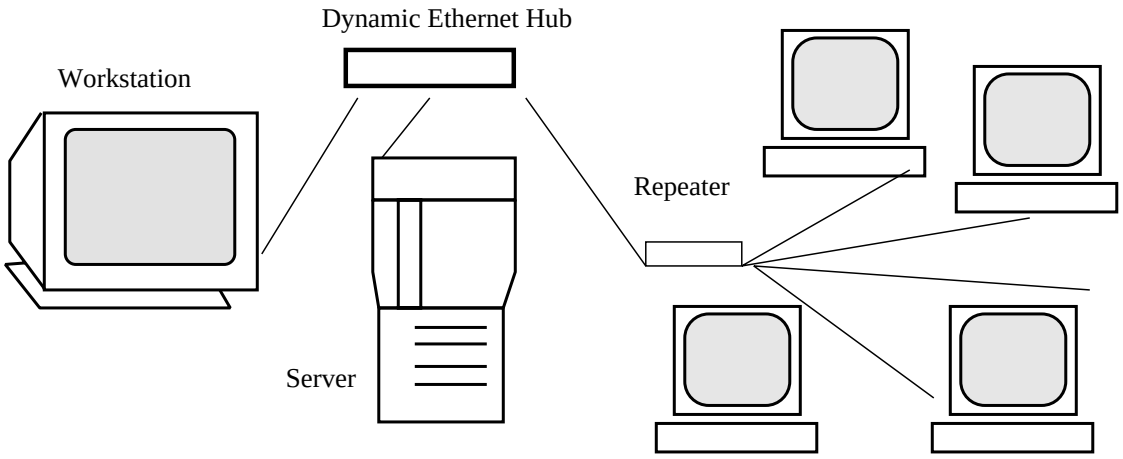

Figure 5: A dynamic ethernet hub in our simple network.

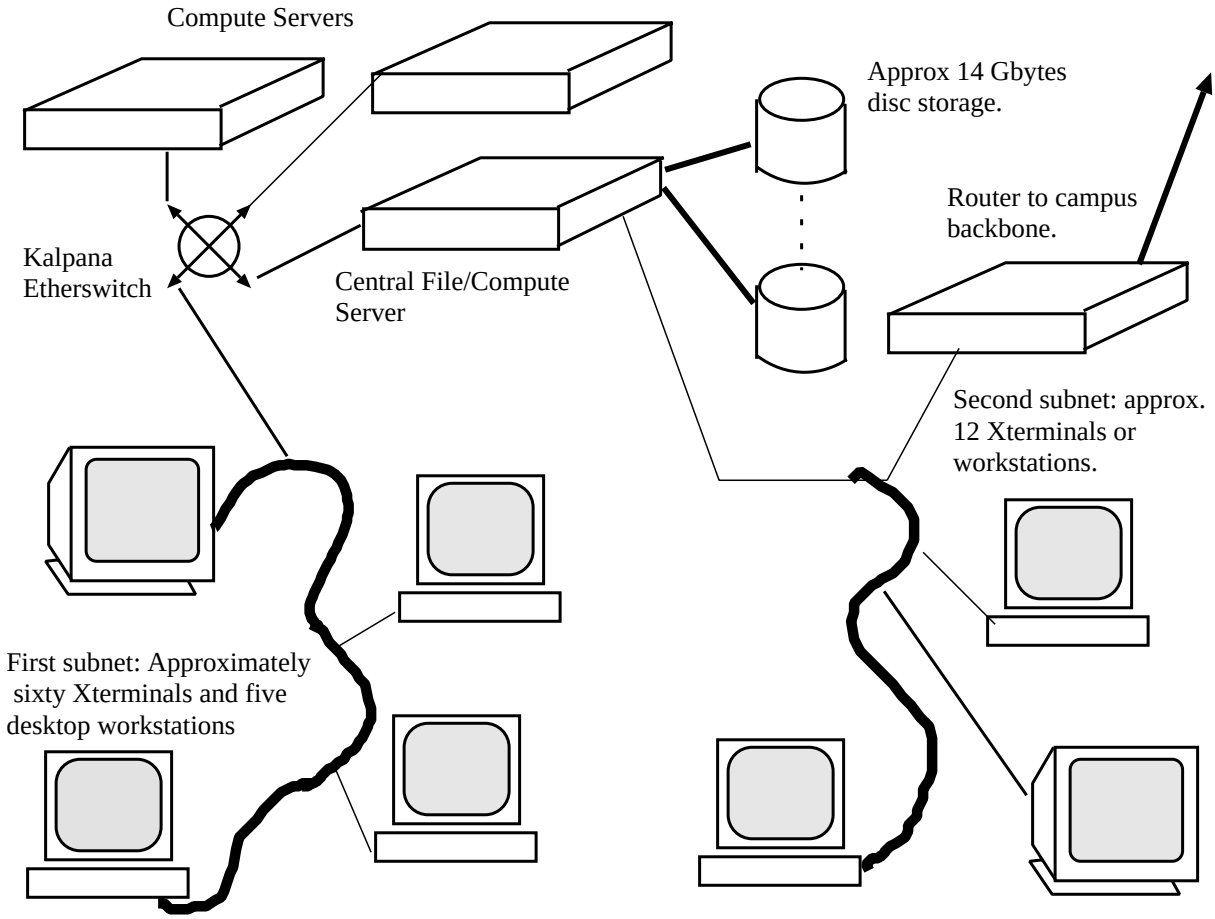

Figure 6: The Computer Science Network, summarised.

interface to our central fileserver with approximately half of the workstations on each subnet did, indeed, cut our network loads in half.

The introduction of two 30 – Xterminal student labs increased our ethernet loads considerably. As well as this we began converting our slower workstations to Xterminals. Suddenly our fileserver and compute servers were working very hard, and the ethernet load on our "Xterminal" subnet averaged 15% and more. Because of the layout of our ethernets, both laboratories were on one subnet.

To cure this situation a dynamic ethernet hub was procured and installed on the Xterminal subnet. Connections were made to one of our compute servers, one port of our central fileserver and our sixty–odd student Xterminals were connected to the third port. This layout was determined, as much as anything else, by the topology of our network.

The particular switching hub we used is a Kalpana EtherSwitch$^{(TM)}$. Our model has five ports, and packets can be routed between any two pairs of ports simultaneously. The routing happens as soon as the first six bytes of an ethernet packet (which contain the destination address) have been received, called "on the fly" switching by Kalpana. Buffering is provided for when two ports receive a packet directed to the same destination. This style of dynamic hub is is called a cross – point switch. Other styles of dynamic ethernet hub use a "fat pipe" (a high – speed local network) or a high – speed backplane to implement their dynamic packet routing. Packets may be switched either on the fly, or received, buffered and then routed (ie, like a traditional router). In this latter case, transit time across the router can be two milliseconds or more, as the packet is received, then retransmitted.

Our switching hub makes a considerable difference to ethernet contention. Here are some network statistics reported by netstat from a workstation on a quiet ethernet using spray to send 10000 bytes to another workstation. Simultaneously the other workstation is spraying 10000 bytes

at it:

```
   input   (le0)      output
packets errs  packets errs  colls
144       0       81      0      3
140       0       86      0      8
139       0       119     0      0
40        0       57      0      8
139       0       728     0      17
1237      0       245     0      9
114       0       399     0      33
40        0       17      0      0
1         0       1       0      0
1         0       1       0      0
1         0       1       0      0
```

Notice the collisions reported in the last column. Here is the same experiment, only this time both machines have their own port on the switching hub:

```
    input   (le0)      output
packets errs  packets errs  colls
10      0     11      0          0
8       0     11      0          0
8       0     11      0          0
10      0     12      0          0
16      0     14      0          0
56      0     57      0          0
287     0     1195    0          0
917     0     13      0          0
9       0     10      0          0
```

The isolation provided by the switching hub cuts down the collisions to zero. Both these experiments were made on a quiet ethernet - what made the difference was the presence of the switching hub.

5 Analysis

Using just three of the five ports on our switching hub resulted in a drop in our average ethernet load from 15% to 5% — a threefold drop. What caused this improvement? First, let's consider the characteristics of the various types of nodes on our network of servers, clients and Xterminals:

- The Xterminals are mainly passive: they send small amounts of data and receive large amounts of data.
- The compute servers both send and receive large amounts of data to and from the central file server, and to the Xterminals.
- The central file server both sends and receives large amounts of NFS traffic. It also acts as a compute server.

Hence, X clients on compute servers communicating with Xterminals usually cause an associated burst of filesharing traffic from the fileserver(s). Figure 7 shows the close correlation between filesharing traffic and Xtraffic resulting from a typical user command: in this case the author displaying the postscript file from a draft version of this paper on his Xterminal. The filesharing traffic results from loading the viewer program and the file, while the X traffic is the program rendering the postscript file on to the terminal. I deliberately chose a workstation on the same ethernet as my terminal for this experiment.

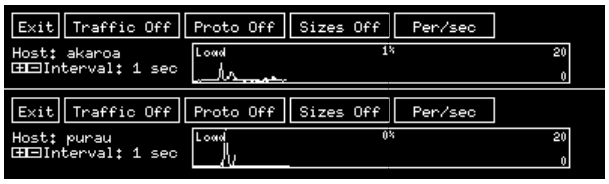

Figure 7: Xtr, an ethernet traffic monitor, reporting the X protocol packets arriving at an Xterminal (top trace) and the NFS packets arriving at the compute server which sent the X packets (bottom trace).

As can be seen, the peak of the X traffic is closely correlated to that of the filesharing traffic, giving a high likelihood of collisions, and this is also borne out by running netstat, which reports statistics from the ethernet interface:

```
    input   (le0)     output
packets errs  packets errs  colls
3       0     1       0     0
3       0     1       0     0
4       0     3       0     0
13      0     9       0     0
1       0     1       0     0
3       0     2       0     0
166     0     63      0     5 ***
58      0     50      0     0
196     0     70      0     0
218     0     49      0     0
136     0     42      0     0
70      0     58      0     0
13      0     3       0     0
2       0     3       0     0
```

The last column reports collisions, and normally is zero because this workstation is on a moderately – utilized ethernet. However, five collisions are reported in line seven, due to filesharing traffic arriving at the workstation at the same time as it is trying to send X traffic to my Xterminal.

A statistical example of the affects of this correlation between X – protocol traffic and filesharing traffic from our server is shown in this comparisan of cumulative network statistics from two of our compute servers. Both of these servers are similarly configured, and provide sessions to large numbers of Xterminals, and use files from our central fileserver. The first compute server has its own port on our dynamic hub:

```
    input   (le0)     output
packets errs   packets  errs  colls
16182339 71    19413032 0     30613
```

Collisions total 0.85% of input and output traffic. This low figure indicates good network response.

The second server shares the subnet which has most of our Xterminals:

```
    input   (le0)     output
packets errs  packets errs  colls
17241365 43   18754759 232 4796730
```

Here collisions total 13% of input and output traffic, which shows just how severe the effects of filesharing traffic competing with X client traffic can be. This is a server with a problem. Giving this server what amounts to a "clear" ethernet, which is to say a dedicated port on our switching hub, would result in a more than tenfold drop in collisions.

The big improvement for our network has been that separating Xterminal traffic from filesharing traffic has produced a considerable increase in network availability, and consequent collision reduction, for our file and compute servers.

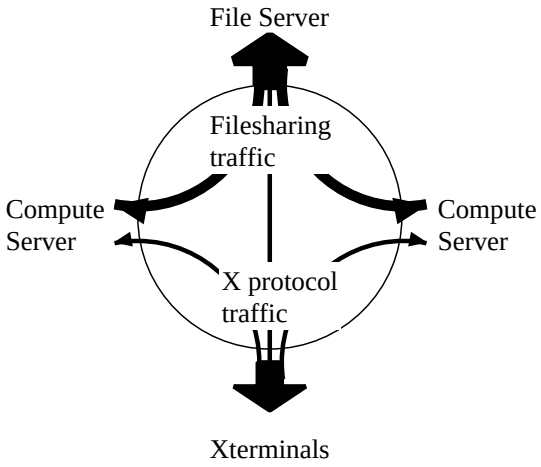

Figure 8: Traffic flows through our switching hub. Thicker lines show heavier traffic.

6 Sensible switching

There are a number of rules for getting the best out of a switched ethernet, or any similarly subnetted ethernet:

1. Identify the main sources and destinations of traffic flow.
2. Categorise the traffic. Is one type of traffic associated with another?
3. Try and use subnetting to separate traffic flows, especially when one flow is correlated with another. In our case we separated X–protocol traffic from the assocated NFS filesharing traffic, see figure 8 for a diagram of the traffic flows through our switching hub.

Some cases of network overload are not very amenable to solution, and all that can be done is to try and minimize the affects on other network users. For example, discless workstations send and receive moderate amounts of filesharing traffic, due to their having swap partitions remotely mounted. However, if the workstation begins paging excessively it can overload its local subnet as well as associated file servers. The only way to fix the overload is to stop the workstation paging, usually by add more memory.

However, for us the use of a dynamically switching ethernet hub has produced a high performance network at a low cost in both financial and administrative terms.

References

[1] A. J. E. Dale. Tuning networked unix systems. In *Proc. of UniForum NZ Conference '93*, pages 9.1–12, Masterton, May 1993.

[2] Mike Loukides. *System Performance Tuning.* Nutshell Handbooks. O'Reilly and Associates Inc, 1990.

[3] Brian L Wong, Pat Shuff, and Hal L Stern. *Configuration and Capacity Planning for Sun Servers.* Sun Microsystems, USA, January 1992.